STPC
PCMCIA Driver Writer's Guide

Issue 1.0

February 22, 2000

STMicroelectronics

1. INTRODUCTION

STPC Cardbus Controller support both PCMCIA (16 bits) and Cardbus (32 bits) cards. The STPC Cardbus Controller is implemented in the PCI bus as a PCI to PCMCIA or PCI to Cardbus bridge.

2. PCMCIA SPECIFICS

The basic PCMCIA design is based on the M82365 controller with few differences in the power control logic. The power control logic register can be found in the PCMCIA register documentation. Due to the PCI implementation, some limitations on resources allocations have been added to the original M82365 design :

- Controller and card interrupt line are fixed to PCI IRQ#A and programmed via the PCI configuration space like a standard PCI device.
- PCMCIA memory and I/O windows are accessible through the PCI memory or I/O space. 2 address ranges are available and programmable via the PCI configuration space like a standard PCI device. PCI address ranges have to be programmed to be compatible with PCMCIA windows programming.

3. THE CARDBUS IS BASED ON YENTA SPECIFICATIONS.

An additionnal PCI configuration space register allows the selection of either PCMCIA or Cardbus mode and have to be programmed by software to reach the format of the inserted PC-CARD.

4. POINTS TO BE TAKEN CARE OF :-

4.1. DYNAMIC MODE CHANGE

There is a mode register in PCI Configuration space of the Cardbus Controller. The register is at offset 0x58h and specifies the mode of the controller ,whether it is in 16-bit or 32-bit. So the mode needs to be dynamically changed as soon as acard is inserted or removed.

4.2. BIOS INITIALISATION

The BIOS should initialize the controller in Cardbus (32 bit) mode or else the controller should be in Cardbus mode just before windows launches. This is because Window does the full enumeration of the Cardbus bridge and tries to enumerate the Cardbus bus. If the controller is in PCMCIA mode the controller will hang which implies that Windows will hang.

4.3. INTERRUPT PROBLEMS

Use of GPIO to remove the interrupt problem . The problem is on pin 16 of the socket i.e. the IREQ# pin which is active low. When certain cards are inserted they assert the signal low which means interrupts start coming for no reason. This keeps on happening until we apply power to card after which it starts driving the line. During this period, interrupts keep coming which hangs the system. The interrupts need to be masked off until power is applied. The GPIO ping needs to be used for this operation. Connect the GPIO pin with the IREQ# as;

IREQ# = IREQ# || GPIO[7]

Initially, keep GPIO[7] as 1 so the final IREQ# signal is 1 and interrupts don't arrive after the power is applied and allow it to stabilise making GPIO[7] '0' which means the actual IREQ# signals start coming from the card.

4.4. IO MAPPED ADDRESS

There is a register at offset 44h of PCI Configuration space of the card which is programmed with the IO address and used to access the PCCard 16 registers .Those register are only IO mapped and are not memory mapped.

The Cardbus register specified at offset 10h of the PCI Configuration space is memory mapped.

4.5. POWER LOGIC REGISTER PROGRAMMING

The power logic for 16bit cards is as follows . In the register at index 02h the first four bits are the power logic bits. They are interpreted as in Table 4-1.

Table 4-1. Power Logic programming

Bit3	Bit2	Bit1	Bit0	VCC	VPP
Vcc3en	Vcc5en	VccVpp	VppPgm		
1	0	1	0	3.3	3.3
1	0	0	1	3.3	12.0
0	1	1	0	5.0	5.0
0	1	0	1	5.0	12.0
0	0	X	X	0.0	0.0

4.6. MEMORY/IO WINDOW SETUP

There are two places where the base and limit settings for memory/IO windows on 16 bit cards need to be done.

One is in the PCMCIA registers namely IO/Memory base register and IO/Memory end register, also in the PCI configuration register Memory/IO base register (1 or 2) and Memory/ IO limit register (1 or 2). ***This limit our memory windows by 2 , but we can emulate 3 windows in 2 in case the windows are contigous .***

For example we want a memory window at location D1000h for a limit of 8k . We will program;

- System Memory Map Start Low Address to D1h
- System Memory Map Start High Address to 0h
- System Memory Map End Low Address to D2h
- System Memory Map End High Address to 0h

and also in the PCI Configuration space;

- at offset 1Ch/24 BAR0/1 Memory Base D1000h
- at offset 20h/28 BAR0/1 Memory Limit D2000h

4.7. CONTROLLER AND CARD INTERRUPT ROUTING

Both controller and card interrupts are routed to PCI IRQ#A what ever value is programmed in the Interrupt Controller Register (Index 003h and 005h). Therefore both controller and card need to share the same IRQ and the card driver must support a shared interrupt line.